Claims

1.   A method for handling data packets, said data packets belonging to a set of data packets, said method comprising the steps of:
- capturing (300, 600) data packets, and
- processing (302, 608) captured data packets,
characterized in that it further comprises the step of:
- accepting (301) a captured data packet for processing or declining (301) a captured data packet from processing based on said captured data packet and data packets captured prior to said captured data packet.

2.   A method according to claim 1, **characterized** in that in that it further comprises the steps of:
- when processing at least one captured data packet, determining (304, 610) a modification command affecting at least said at least one captured data packet, and
- maintaining (305, 610) a list of modification commands, said list enabling modification of captured data packets.

3.   A method according to claim 2, **characterized** in that it further comprises the steps of:
- modifying (306, 612) captured data packets based on said list of modification commands, and
- releasing (308, 614) modified captured data packets.

4.   A method according to claim 1, **characterized** in that captured data packets that are declined from processing are discarded (408, 604).

5.   A method according to claim 1, **characterized** in that captured data packets that are declined from processing are delayed (410, 603).

6.   A method according to claim 2, **characterized** in that it further comprises the steps of:
- declining (404) a captured data packet from processing, if said captured data packet is already processed and modification commands induced by said captured data packet are already determined and maintained in said list of modification commands,
- modifying (306, 612) said captured data packet based on said list of modification commands, and
- releasing (308, 614) the modified captured data packet.

7. A method according to claim 1, **characterized** in that in that in that it further comprises the steps of:

- declining (404) a captured data packet from processing, if said captured data packet is already processed, and

5   - releasing (308) the captured data packet.

8. A method according to claim 1, **characterized** in that said set of data packets has at least partly hierarchical structure, said method further comprising the step of:

- accepting (301, 411, 414, 422, 430) data packets for processing in the order specified by said at least partly hierarchical structure.

10 9. A method according to claim 8, **characterized** in

- accepting (411) a captured data packet for processing, if data packet(s) immediately preceding said captured data packet in said at least partly hierarchical structure is(are) already processed.

10. A method according to claim 8, **characterized** in

15 - declining (404) a captured data packet from processing, if a data packet immediately preceding said captured data packet in said at least partly hierarchical structure is not yet captured.

11. A method according to claim 10, **characterized** in

- delaying (410, 603) data packets declined from processing,

20 - accepting (414) a delayed data packet for processing and processing (302) said delayed data packet, if data packet(s) immediately preceding said delayed data packet in said at least partly hierarchical structure is(are) already processed, and

- releasing (308) the delayed data packet.

12. A method according to claim 10, **characterized** in

25 - delaying (410, 603) data packets declined from processing,

- accepting (301, 411, 414, 422, 430) a captured data packet for processing,

- accepting (414) delayed data packet(s) for processing, if data packet(s) immediately preceding said delayed data packet(s) in said at least partly hierarchical structure is(are) accepted for processing,

30 - processing (418, 424) said delayed data packet(s) together with said captured data packet, and

- releasing (308) the delayed and the captured data packets.

13. A method according to claim 8, **characterized** in that said at least partly hierarchical structure is a sequence of data packets.

14. A method according to claim 8, **characterized** in that said at least partly hierarchical structure is a hierarchically structured tree.

15. A method according to claim 1, **characterized** in that said data packets form a plurality of groups of data packets, said method further comprising the steps of:

5 - accepting (301) the captured data packets for processing in groups.

16. A method according to claim 15, **characterized** in
- declining (404) a captured data packet belonging to a first group of data packets from processing, if all other data packets belonging to said first group of data packets are not yet captured, and delaying (410) said captured data packet, and
10 - processing (424) a captured data packet belonging to a first group of data packets together with delayed data packets belonging to said first group of data packets, if said captured data packet belonging to said first group and said delayed data packets belonging to said first group form a full first group of data packets.

17. A method according to claim 15, **characterized** in that said plurality of groups
15 of data packets further belong to a set of groups having at least partly hierarchical structure, and in that said method further comprises the step of:
- processing (302) said groups of captured data packets in the order specified by said at least partly hierarchical structure.

18. A method according to claim 2, **characterized** in that said data packets
20 belonging to a set of data packets are first handled (800) in a first node of a cluster of network elements and said list of modification commands is maintained in said first node, and in that said method further comprises the step of:
- transmitting (804) said list of modification commands from said first node to a second node of said cluster of network elements.

25 19. A method according to claim 18, **characterized** in that it further comprises the step of:
- after said transmission of said list, handling (806) said set of data packets in said second node.

20. A method according to claim 19, **characterized** in that it further comprises the
30 steps of:
- when beginning to handle said data packets, storing in said first node in a connection data structure an entry representing said set of data packet, and
- before handling said set of data packets in said second node, transmitting said entry from said first node to said second node.

21.   A method according to claim 1, **characterized** in that it further comprises the step of:

- defining (700) a plurality of first pieces of information, which are to be replaced in the captured data packets with a plurality of corresponding second pieces of information,

and in that in processing captured data packets,

- said first pieces of information are searched for (702) , and

- if a first piece of information is found, at least one modification command specifying at least the replacement of said first piece of information with a corresponding second piece of information is determined (704).

22.   A method according to claim 21, **characterized** in that if the length of said first piece of information is different from the length of said corresponding second piece of information and if said first piece of information is found in payload of data packet(s), said modification command comprises instructions for changing value of at least one header field in a data packet.

23.   A method according to claim 2, **characterized** in that a modification command comprises

- a first identifier indicating the beginning of a first piece of information in the original captured data packets, the first piece of information being subject to be replaced by a second piece of information,

- the length of the first piece of information, and

- the second piece of information.

24.   A method according to claim 23, **characterized** in that a modification command further comprises

- a second identifier indicating the beginning of the second piece of information in the modified captured data packets,

- an offset between a third identifier indicating the end of the first piece of information in the original captured data packets and a fourth identifier indicating the end of the second piece of information in the modified captured data packets, and

- the length of the second piece of information.

25.   A method according to claim 1, **characterized** in that said data packets contain information fragments belonging to a sequence of information fragments, said method further comprising the steps of:

- processing (302, 418, 424, 432, 608) the information fragments of the captured data packets in the order specified by said sequence.

26. A method according to claim 25, **characterized** in that each information fragment of said sequence is processed only once.

27. A method according to claim 25, **characterized** in that said method further comprises the steps of:
- declining (404) a captured data packet from processing, if a data packet containing the information fragment immediately preceding the information fragments of said captured data packet in said sequence is not yet captured, and
- accepting (411) a captured data packet for processing, if a data packet containing the information fragment immediately preceding the information fragments of said captured data packet in said sequence is already processed.

28. A method according to claim 25, **characterized** in that said sequence of information fragments is a sequence of octets of data according to Transfer Control Protocol.

29. A software entity (900) for handling data packets, said data packets belonging to a set of data packets, said software entity comprising
- program code means (901) for capturing data packets,
characterized in that it further comprises:
- program code means (904) for accepting a captured data packet for processing or declining a captured data packet from processing based on said captured data packet and data packets captured prior to said captured data packet.

30. A software entity (900) according to claim 29, **characterized** in that it further comprises:
- program code means (908) for maintaining a list of modification commands, said list enabling modification of captured data packets.

31. A software entity (900) according to claim 30, **characterized** in that it further comprises:
- program code means (910) for modifying captured data packets based on said list of modification commands, and
- program code means (912) for releasing modified captured data packets.

32. A software entity (920) for processing data, said software entity being adapted to receive data, and said software entity comprising

- program code means (902) for processing received data,

- program code means (906) for determining a modification command affecting at least received data, as a response to processing said data, and

- said software entity being adapted to output said modification command.

5   33.   A computer program comprising program code for performing all the steps of Claim 1 when said program is run on a computer.

34.   A computer program product comprising program code means stored on a computer readable medium for performing the method of Claim 1 when said program product is run on a computer.

10   35.   A network element (952) for handling data packets, said data packets belonging to a set of data packets, said network element comprising
- means (954) for capturing data packets,
characterized in that it further comprises:
- means (958) for accepting a captured data packet for processing or declining a
15   captured data packet from processing based on said captured data packet and data packets captured prior to said captured data packet.

36.   A network element (952) according to claim 35, **characterized** in that it further comprises
- means (956) for processing a captured data packet, and
20   - means (960) for determining a modification command affecting at least one captured data packet, as a response to processing said at least one captured data packet.

37.   A network element cluster (950) for handling data packets, said data packets belonging to a set of data packets, at least one node (952) of said network element
25   cluster comprising
- means (954) for capturing data packets,
characterized in that it further comprises:
- means (958) for accepting a captured data packet for processing or declining a captured data packet from processing based on said captured data packet and data
30   packets captured prior to said captured data packet.

38.   A network element cluster (950) according to claim 37, **characterized** in that said at least one node (952) further comprises
- means (962) for maintaining a list of modification commands, said list enabling modification of captured data packets, and

- means (964) for transmitting said list of modification commands from said node to another node of said cluster of network elements.